

BOD
BOD

BOD

BOD
BOD

BOD
BOD

BOD
BOD

BOD
BOD

BOD
BOD

BOD BUG

**BUG
BYE**

**BYP
CAN**

CAN
CAN

**CAN
CAN**

**CAN
CHE**

CHE
CHE

CHE

CLU

CLU

CLU

CLU
CLU

CLU
CLU

CLU
CLU

CLU

CLU

CLU
CLU

CLU
CLU

CLU
CLU

CLU
CLU

CLU

CLU

CLU

OP
VO

.....

[illegible]

```
0001 0 MODULE OPC$OPCOMOLD (
0002 0     LANGUAGE (BLISS32),
0003 0     IDENT = 'V04-000'
0004 0 ) =
0005 0
0006 0 *****
0007 0 *
0008 0 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 0 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 0 *  ALL RIGHTS RESERVED.
0011 0 *
0012 0 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 0 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 0 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 0 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 0 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 0 *  TRANSFERRED.
0018 0 *
0019 0 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 0 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 0 *  CORPORATION.
0022 0 *
0023 0 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 0 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 0 *
0026 0 *
0027 0 *****
0028 0
0029 0 ++
0030 0 FACILITY:
0031 0
0032 0     OPCOM
0033 0
0034 0 ABSTRACT:
0035 0
0036 0     This module contains the old format message handlers.
0037 0     These routines merely reformat the request into the
0038 0     new format, and call the correct handler to service
0039 0     the request.
0040 0
0041 0 Environment:
0042 0
0043 0     VAX/VMS operating system.
0044 0
0045 0 Author:
0046 0
0047 0     Steven T. Jeffreys
0048 0
0049 0 Creation date:
0050 0
0051 0     March 10, 1981
0052 0
0053 0 Revision history:
0054 0
0055 0     V03-002 CWH3001      CW Hobbs      30-Jul-1983
0056 0     Various and sundry things to make OPCOM distributed
0057 0     across the cluster.
```



```

58 0058 0 |
59 0059 0 | V03-001 STJ3033 Steven T. Jeffreys, 05-Oct-1982
60 0060 0 | Make all operators permanent by default.
61 0061 0 |
62 0062 0 | V02-002 STJ0223 Steven T. Jeffreys, 17-Feb-1982
63 0063 0 | Make all operators temporary by default.
64 0064 0 |
65 0065 0 |
66 0066 0 | --
67 0067 0 |
68 0068 1 BEGIN ! Start of OPCOMOLD
69 0069 1
70 0070 1 LIBRARY 'SYSS$LIBRARY:LIB.L32';
71 0071 1 LIBRARY 'LIB$:OPCOMLIB';
72 0072 1
73 0073 1 FORWARD ROUTINE
74 0074 1
75 0075 1 Various message handlers for old format messages.
76 0076 1
77 0077 1 CNCL_HANDLER : NOVALUE, Cancel handler
78 0078 1 LOGI_HANDLER : NOVALUE, Init logfile message handler
79 0079 1 RPLY_HANDLER : NOVALUE, Reply handler
80 0080 1 RQST_HANDLER : NOVALUE, Request handler
81 0081 1 SECU_HANDLER : NOVALUE, Security handler
82 0082 1 STS_HANDLER : NOVALUE, Status handler
83 0083 1 TERME_HANDLER : NOVALUE, Enable operator message handler
84 0084 1

```

```

86 0085 1 GLOBAL ROUTINE CNCL_HANDLER (BUFFER_DESC) : NOVALUE =
87 0086 1
88 0087 1 ++
89 0088 1 Functional description:
90 0089 1
91 0090 1 This routine is the handler for all CNCL messages received by OPCOM.
92 0091 1 This message is in the old format, and must be converted to the new
93 0092 1 format before it can be processed. Once this is done, the new format
94 0093 1 message handler is called to process the reformatted request.
95 0094 1
96 0095 1 Input:
97 0096 1
98 0097 1 BUFFER_DESC : The address of a quadword buffer descriptor that
99 0098 1 describes the buffer containing the message.
100 0099 1
101 0100 1 Implicit Input:
102 0101 1
103 0102 1 None.
104 0103 1
105 0104 1 Output:
106 0105 1
107 0106 1 None.
108 0107 1
109 0108 1 Implicit output:
110 0109 1
111 0110 1 Some accounting data will be updated
112 0111 1 to reflect the receipt of the message.
113 0112 1
114 0113 1 Side effects:
115 0114 1
116 0115 1 None.
117 0116 1
118 0117 1 Routine value:
119 0118 1
120 0119 1 None.
121 0120 1 --
122 0121 1
123 0122 2 BEGIN ! Start of CNCL_HANDLER
124 0123 2
125 0124 2 MAP
126 0125 2
127 0126 2 BUFFER_DESC : $ref_bblock;
128 0127 2
129 0128 2 EXTERNAL
130 0129 2 GLOBAL_STATUS : BITVECTOR; ! Global status flags
131 0130 2
132 0131 2 EXTERNAL ROUTINE
133 0132 2 CANCEL_HANDLER : NOVALUE; ! New format msg handler
134 0133 2
135 0134 2 LOCAL
136 0135 2
137 0136 2 OLD_MSG : $ref_bblock, ! Pointer to start of old message
138 0137 2 NEW_MSG : $ref_bblock, ! Pointer to start of new message
139 0138 2 REFORMAT_BUFFER : $bblock [OP($K_MAXREAD), ! Buffer to hold the reformatted message
140 0139 2 REFORMAT_DESC : $desc_block; ! Descriptor for the REFORMAT_BUFFER
141 0140 2
142 0141 2 !
```

```
143 0142 2 1 Make sure the message is big enough. If not, it
144 0143 2 1 cannot possibly be a valid message, and OPCOM will
145 0144 2 1 simply ignore it.
146 0145 2 1
147 0146 2 1 IF .BUFFER_DESC [DSC$W_LENGTH] LSS (OPC$K_COMHDRSIZ + 8)
148 0147 2 1 THEN
149 0148 2 1 RETURN;
150 0149 2 1
151 0150 2 1 Copy the common header provided by $SENDPR to the new buffer
152 0151 2 1
153 0152 2 1 CHSMOVE (OPC$K_COMHDRSIZ, .BUFFER_DESC [DSC$A_POINTER], REFORMAT_BUFFER);
154 0153 2 1
155 0154 2 1 Zero the remainder of the REFORMAT_BUFFER.
156 0155 2 1
157 0156 2 1 CH$FILL (0, (OPC$K_MAXREAD - OPC$K_COMHDRSIZ), (REFORMAT_BUFFER + OPC$K_COMHDRSIZ));
158 0157 2 1
159 0158 2 1 Move the old message fields into their corresponding places in the new message format.
160 0159 2 1
161 0160 2 1 OLD_MSG = .BUFFER_DESC [DSC$A_POINTER] + OPC$K_COMHDRSIZ;
162 0161 2 1 NEW_MSG = REFORMAT_BUFFER + OPC$K_COMHDRSIZ;
163 0162 2 1 NEW_MSG [OPC$B_RQSTCODE] = .OLD_MSG [OPC$B_MS_TYPE];
164 0163 2 1 NEW_MSG [OPC$B_SCOPE] = OPC$K_SYSTEM;
165 0164 2 1 NEW_MSG [OPC$L_ATTNUMASK1] = .OLD_MSG [$BYTEOFFSET (OPC$B_MS_TARGET), 0, 24, 0];
166 0165 2 1 NEW_MSG [OPC$L_RQSTID] = .OLD_MSG [OPC$L_MS_RQSTID];
167 0166 2 1
168 0167 2 1
169 0168 2 1 Create a descriptor for the reformatted message.
170 0169 2 1
171 0170 2 1 REFORMAT_DESC [DSC$W_LENGTH] = OPC$K_COMHDRSIZ + OPC$K_HDR_SIZE;
172 0171 2 1 REFORMAT_DESC [DSC$B_DTYPE] = 0;
173 0172 2 1 REFORMAT_DESC [DSC$B_CLASS] = 0;
174 0173 2 1 REFORMAT_DESC [DSC$A_POINTER] = REFORMAT_BUFFER;
175 0174 2 1
176 0175 2 1 Call the new-message handler to finish processing the message.
177 0176 2 1
178 0177 2 1 GLOBAL_STATUS [GBLSTS_K_OLD_FORMAT_MSG] = TRUE; ! Mark this as an old format msg
179 0178 2 1 CANCEL_HANDLER (REFORMAT_DESC);
180 0179 2 1 GLOBAL_STATUS [GBLSTS_K_OLD_FORMAT_MSG] = FALSE;
181 0180 2 1
182 0181 2 1
183 0182 2 1
184 0183 2 1
185 0184 2 1
186 0185 1 END; ! End of CNCL_HANDLER
```

```
.TITLE OPCSOPCOMOLD
.IDENT \V04-000\
```

```
.EXTRN GLOBAL_STATUS, CANCEL_HANDLER
```

```
.PSECT $CODE$, NOWRT, 2
```

```
.ENTRY CNCL_HANDLER, Save R2, R3, R4, R5, R6
MOVAB -2568(SP), SP
MOVL BUFFER_DESC, R6
CMPW (R6), #46
```

```
SE 007C 00000
56 F5F8 CE 9E 00002
2E 04 AC D0 00007
66 B1 0000B
```

```
: 0085
: 0146
:
```


09DA	BF	08	AE	04	B6	45	1F	0000E	BLSSU	1\$:	0153
			00		6E	26	28	00010	MOVC3	#38, @4(R6), REFORMAT_BUFFER	:	0158
						00	2C	00016	MOVC5	#C, (SP), #0, #2522, REFORMAT_BUFFER+38	:	
			51	04	A6	2E	AE	0001D			:	
					50	2E	AE	0001F	ADDL3	#38, 4(R6), OLD_MSG	:	0163
					60		AE	00024	MOVAB	REFORMAT_BUFFER+38, NEW_MSG	:	0164
					60		61	90	MOVAB	(OLD_MSG), (NEW_MSG)	:	0165
					A0		01	90	MOVAB	#1, T(NEW_MSG)	:	0166
0A	A0	01	A1	01	18		00	EF	EXTZV	#0, #24, T(OLD_MSG), 10(NEW_MSG)	:	0167
					12	04	A1	D0	MOVL	4(OLD_MSG), 18(NEW_MSG)	:	0168
					6E	40	8F	9A	MOVZBL	#64, REFORMAT_DESC	:	0173
					AE	08	AE	9E	MOVAB	REFORMAT_BUFFER, REFORMAT_DESC+4	:	0176
				0000G	CF		10	88	BISB2	#16, GLOBAL_STATUS	:	0181
							5E	DD	PUSHL	SP	:	0182
				0000G	CF		01	FB	CALLS	#1, CANCEL_HANDLER	:	
				0000G	CF		10	8A	BICB2	#16, GLOBAL_STATUS	:	0183
							04	00055	RET	1\$:	:	0185

; Routine Size: 86 bytes, Routine Base: \$CODE\$ + 0000

```
188 0186 1 GLOBAL ROUTINE LOGI_HANDLER (BUFFER_DESC) : NOVALUE =
189 0187 1
190 0188 1 ++
191 0189 1 Functional description:
192 0190 1
193 0191 1 This routine is the handler for all LOGI messages received by OPCOM.
194 0192 1 This message is in the old format, and must be converted to the new
195 0193 1 format before it can be processed. Once this is done, the new format
196 0194 1 message handler is called to process the reformatted request.
197 0195 1
198 0196 1 Input:
199 0197 1
200 0198 1 BUFFER_DESC : The address of a quadword buffer descriptor that
201 0199 1 describes the buffer containing the message.
202 0200 1
203 0201 1 Implicit Input:
204 0202 1
205 0203 1 None.
206 0204 1
207 0205 1 Output:
208 0206 1
209 0207 1 None.
210 0208 1
211 0209 1 Implicit output:
212 0210 1
213 0211 1 Some accounting data will be updated
214 0212 1 to reflect the receipt of the message.
215 0213 1
216 0214 1 Side effects:
217 0215 1
218 0216 1 None.
219 0217 1
220 0218 1 Routine value:
221 0219 1
222 0220 1 None.
223 0221 1 --
224 0222 1
225 0223 2 BEGIN ! Start of LOGI_HANDLER
226 0224 2
227 0225 2 MAP
228 0226 2
229 0227 2 BUFFER_DESC : $ref_bblock;
230 0228 2
231 0229 2 EXTERNAL
232 0230 2 GLOBAL_STATUS : BITVECTOR, ! Global status flags
233 0231 2 DEVICE_FAO : $bblock; ! FAO control string
234 0232 2
235 0233 2 EXTERNAL ROUTINE
236 0234 2 LOGFILE_HANDLER : NOVALUE; ! New format msg handler
237 0235 2
238 0236 2 LOCAL
239 0237 2
240 0238 2 OLD_MSG : $ref_bblock, ! Pointer to start of old message
241 0239 2 NEW_MSG : $ref_bblock, ! Pointer to start of new message
242 0240 2 OUT_LENGTH : WORD, ! Length of formatted operator device name
243 0241 2 DEV_DESC : $desc_block, ! Operator device name descriptor
244 0242 2 REFORMAT_BUFFER : $bblock [OP($K_MAXREAD), ! Buffer to hold the reformatted message
```



```
245      REFORMAT_DESC : $desc_block;                ! Descriptor for the REFORMAT_BUFFER
246
247
248      Make sure the message is big enough. If not, it
249      cannot possibly be a valid message, and OPCOM will
250      simply ignore it.
251
252      IF .BUFFER_DESC [DSC$W_LENGTH] LSS (OPC$K_COMHDRSIZ + 13)
253      THEN
254          RETURN;
255
256      Copy the commom header provided by $SENDPR to the new buffer
257      CH$MOVE (OPC$K_COMHDRSIZ, .BUFFER_DESC [DSC$A_POINTER], REFORMAT_BUFFER);
258
259      Zero the remainder of the REFORMAT_BUFFER.
260      CH$FILL (0, (OPC$K_MAXREAD - OPC$K_COMHDRSIZ), (REFORMAT_BUFFER + OPC$K_COMHDRSIZ));
261
262      Move the old message fields into their corresponding places in the new message format.
263
264      OLD_MSG = .BUFFER_DESC [DSC$A_POINTER] + OPC$K_COMHDRSIZ;
265      NEW_MSG = REFORMAT_BUFFER + OPC$K_COMHDRSIZ;
266      NEW_MSG [OPC$B_RQSTCODE] = .OLD_MSG [OPC$B_MS_TYPE];                ! Set request type
267      NEW_MSG [OPC$B_SCOPE] = OPC$K_SYSTEM;                            ! Force SYSTEM request
268      IF .OLD_MSG [OPC$L_MS_RQSTID] EQL 0
269      THEN
270          $bblock [NEW_MSG [OPC$L_RQ_OPTIONS], OPC$V_INITLOG] = TRUE ! INITLOG function
271      ELSE
272          $bblock [NEW_MSG [OPC$L_RQ_OPTIONS], OPC$V_CLOSELOG] = TRUE;    ! CLOSELOG function
273
274      Build the operator device name from the ASCII device string
275      and the device unit number. Build the FAO OUTBUF descriptor
276      to point to the correct spot within NEW_MSG to save a copy.
277
278      DEV_DESC [0,0,32,0] = 20;                ! Allow for a large device name
279      DEV_DESC [DSC$A_POINTER] = .NEW_MSG + $BYTEOFFSET (OPC$T_LOGFILE_OPR) + 1;
280      $FAO (DEVICE FAO, OUT_LENGTH, DEV_DESC, OLD_MSG [OPC$T_MS_ONAME], .OLD_MSG [OPC$W_MS_OUNIT]);
281      NEW_MSG [$BYTEOFFSET (OPC$T_LOGFILE_OPR),0,8,0] = .OUT_LENGTH;
282
283      Create a descriptor for the reformatted message.
284
285      REFORMAT_DESC [DSC$W_LENGTH] = OPC$K_COMHDRSIZ + OPC$K_HDR_SIZE + .OUT_LENGTH + 1;
286      REFORMAT_DESC [DSC$B_DTYPE] = 0;
287      REFORMAT_DESC [DSC$B_CLASS] = 0;
288      REFORMAT_DESC [DSC$A_POINTER] = REFORMAT_BUFFER;
289
290      Call the new-message handler to finish processing the message.
291
292      GLOBAL STATUS [GBLSTS_K_OLD_FORMAT_MSG] = TRUE;                ! Mark this as an old format msg
293      LOGFILE_HANDLER (REFORMAT_DESC);
294      GLOBAL STATUS [GBLSTS_K_OLD_FORMAT_MSG] = FALSE;
```

: 302
: 303 0300 2
0301 1 END;

! End of LOGI_HANDLER

09DA	8F	OC	AE	04	B6	5E	F5EC	007C	00000	.EXTRN	DEVICE FAO, LOGFILE_HANDLER	
			00		6E	56	04	CE	9E	.EXTRN	SYSSFAO	
			50	04	A6	33		AC	00	.ENTRY	LOGI_HANDLER, Save R2,R3,R4,R5,R6	0186
					52		32	66	B1	MOVAB	-2580(SP), SP	0250
					62			74	1F	MOVL	BUFFER_DESC, R6	
				01	A2			26	2B	CMPW	(R6), #51	
								00	2C	BLSSU	3\$	0257
								AE		MOVC3	#38, #4(R6), REFORMAT_BUFFER	0262
								26	C1	MOVC5	#0, (SP), #0, #2522, REFORMAT_BUFFER+38	
								AE				
								26	9E	ADL3	#38, 4(R6), OLD_MSG	0267
								AE		MOVAB	REFORMAT_BUFFER+38, NEW_MSG	0268
								60	90	MOVB	(OLD_MSG), (NEW_MSG)	0269
								01	90	MOVB	#1, T(NEW_MSG)	0270
								AE		TSTL	4(OLD_MSG)	0271
								06	12	BNEQ	1\$	
				06	A2			01	88	BISB2	#1, 6(NEW_MSG)	0273
								04	11	BRB	2\$	
								02	88	BISB2	#2, 6(NEW_MSG)	0275
				06	A2			14	00	MOVL	#20, DEV_DESC	0281
				FB	AD			AE	9E	MOVAB	27(R2), DEV_DESC+4	0282
				FC	AD			A0	3C	MOVZWL	8(OLD_MSG), --(SP)	0283
					7E			AE	9F	PUSHAB	10(OLD_MSG)	
								AD	9F	PUSHAB	DEV_DESC	
								AE	9F	PUSHAB	OUT_LENGTH	
								CF	9F	PUSHAB	DEVICE FAO	
								05	FB	CALLS	#5, SYSSFAO	
				04	AE	00000000G	00	6E	90	MOVB	OUT_LENGTH, 26(NEW_MSG)	0284
					1A		A2	8F	A1	ADDW3	#65, OUT_LENGTH, REFORMAT_DESC	0289
					6E		6E	AE	B4	CLRW	REFORMAT_DESC+2	0290
								AE	9E	MOVAB	REFORMAT_BUFFER, REFORMAT_DESC+4	0292
								10	88	BISB2	#16, GLOBAL_STATUS	0297
								AE	9F	PUSHAB	REFORMAT_DESC	0298
								01	FB	CALLS	#1, LOGFILE_HANDLER	
								10	8A	BICB2	#16, GLOBAL_STATUS	0299
								04	00084	RET		0301

; Routine Size: 133 bytes, Routine Base: \$CODE\$ + 0056

```
305 0302 1 GLOBAL ROUTINE RPLY_HANDLER (BUFFER_DESC) : NOVALUE =
306 0303 1
307 0304 1
308 0305 1 **
309 0306 1 Functional description:
310 0307 1
311 0308 1 This routine is the handler for all RPLY messages received by OPCOM.
312 0309 1 This message is in the old format, and must be converted to the new
313 0310 1 format before it can be processed. Once this is done, the new format
314 0311 1 message handler is called to process the reformatted request.
315 0312 1
316 0313 1 Input:
317 0314 1
318 0315 1 BUFFER_DESC : The address of a quadword buffer descriptor that
319 0316 1 describes the buffer containing the message.
320 0317 1
321 0318 1 Implicit Input:
322 0319 1
323 0320 1 None.
324 0321 1
325 0322 1 Output:
326 0323 1
327 0324 1 None.
328 0325 1
329 0326 1 Implicit output:
330 0327 1
331 0328 1 Some accounting data will be updated
332 0329 1 to reflect the receipt of the message.
333 0330 1
334 0331 1 Side effects:
335 0332 1
336 0333 1 None.
337 0334 1
338 0335 1 Routine value:
339 0336 1
340 0337 1 --
341 0338 1
342 0339 2 BEGIN ! Start of RPLY_HANDLER
343 0340 2
344 0341 2 MAP
345 0342 2
346 0343 2 BUFFER_DESC : $ref_bblock;
347 0344 2
348 0345 2 EXTERNAL
349 0346 2 GLOBAL_STATUS : BITVECTOR, ! Global status flags
350 0347 2 DEVICE_FAO : $bblock; ! FAO control string
351 0348 2
352 0349 2 EXTERNAL ROUTINE
353 0350 2 REPLY_HANDLER : NOVALUE; ! New format msg handler
354 0351 2
355 0352 2 LOCAL
356 0353 2
357 0354 2 OLD_MSG_LEN : LONG, ! Length of old message
358 0355 2 NEW_MSG_LEN : LONG, ! Length of new message
359 0356 2 OLD_MSG : $ref_bblock, ! Pointer to start of old message
360 0357 2 NEW_MSG : $ref_bblock, ! Pointer to start of new message
361 0358 2 OUT_LENGTH : WORD, ! Length of formatted operator device name
```



```
362 0359      DEV_DESC      : $desc_block;           ! Operator device name descriptor
363 0360      REFORMAT_BUFFER : $bblock [OPCSK_MAXREAD]; ! Buffer to hold the reformatted message
364 0361      REFORMAT_DESC   : $desc_block;           ! Descriptor for the REFORMAT_BUFFER
365 0362
366 0363
367 0364      ! Make sure the message is big enough. If not, it
368 0365      ! cannot possibly be a valid message, and OPCOM will
369 0366      ! simply ignore it.
370 0367
371 0368      IF .BUFFER_DESC [DSCSW_LENGTH] LSS (OPCSK_COMHDRSZ + 13)
372 0369      THEN
373 0370          RETURN;
374 0371
375 0372      ! Copy the common header provided by $SENDPR to the new buffer
376 0373
377 0374      CHSMOVE (OPCSK_COMHDRSZ, .BUFFER_DESC [DSCSA_POINTER], REFORMAT_BUFFER);
378 0375
379 0376
380 0377      ! Zero the remainder of the REFORMAT_BUFFER.
381 0378
382 0379      CHSFILL (0, (OPCSK_MAXREAD - OPCS_K_COMHDRSZ), (REFORMAT_BUFFER + OPCS_K_COMHDRSZ));
383 0380
384 0381
385 0382      ! Move the old message fields into their corresponding places in the new message format.
386 0383
387 0384
388 0385      OLD_MSG = .BUFFER_DESC [DSCSA_POINTER] + OPCS_K_COMHDRSZ; ! Set pointer to request text
389 0386      NEW_MSG = REFORMAT_BUFFER + OPCS_K_COMHDRSZ;             ! Set pointer to start of new message.
390 0387      NEW_MSG [OPCSB_RQSTCODE] = .OLD_MSG [OPCSB_MS_TYPE];      ! Set message type
391 0388      NEW_MSG [OPCSB_SCOPE] = OPCS_K_SYSTEM;                    ! Force to SYSTEM reply
392 0389      NEW_MSG [OPCSL_RQSTID] = .OLD_MSG [OPCSL_MS_RPLYID];      ! Set reply number
393 0390      NEW_MSG [OPCSL_RQ_OPTIONS] = .OLD_MSG [OPCSQ_MS_STATUS] + OPCS_FACILITY*16; ! Set reply status
394 0391
395 0392      ! Build the operator device name from the ASCII device string
396 0393      ! and the device unit number. Build the FAO OUTBUF descriptor
397 0394      ! to point to the correct spot within NEW_MSG to save a copy.
398 0395
399 0396      DEV_DESC [0,0,32,0] = 20; ! Allow for a large device name
400 0397      DEV_DESC [DSCSA_POINTER] = .NEW_MSG + $BYTEOFFSET (OPCST_REPLY_OPR) + 1;
401 0398      $FAO (DEVICE FAO, OUT_LENGTH, DEV_DESC, OLD_MSG [OPCST_MS_ONAME], .OLD_MSG [OPCSW_MS_OUNIT]);
402 0399      NEW_MSG [$BYTEOFFSET (OPCST_REPLY_OPR), 0, 8, 0] = .OUT_LENGTH;
403 0400      OLD_MSG_LEN = OPCS_K_COMHDRSZ + $BYTEOFFSET (OPCSL_MS_OTEXT);
404 0401      NEW_MSG_LEN = $BYTEOFFSET (OPCST_REPLY_OPR) + .OUT_LENGTH + 1;
405 0402
406 0403      ! Check for the presence of some REPLY text.
407 0404
408 0405      IF .BUFFER_DESC [DSCSW_LENGTH] GTR .OLD_MSG_LEN
409 0406      THEN
410 0407          BEGIN
411 0408              ! There is some reply text present. Copy it to
412 0409              ! the new message buffer.
413 0410
414 0411              CHSMOVE ( (.BUFFER_DESC [DSCSW_LENGTH] - .OLD_MSG_LEN),
415 0412                      OLD_MSG [$BYTEOFFSET (OPCSL_MS_OTEXT), 0, 0, 0],
416 0413                      NEW_MSG [.NEW_MSG_LEN + 2, 0, 0, 0]
417 0414                      );
418 0415
```

```
419 0416 NEW_MSG [.NEW_MSG_LEN,0,16,0] = .BUFFER_DESC [DSCSW_LENGTH] - .OLD_MSG_LEN;
420 0417 NEW_MSG_LEN = .NEW_MSG_LEN + .NEW_MSG [.NEW_MSG_LEN,0,16,0];
421 0418 END;
422 0419 NEW_MSG_LEN = .NEW_MSG_LEN + 2;
423 0420
424 0421
425 0422 Create a descriptor for the reformatted message.
426 0423
427 0424 REFORMAT_DESC [DSCSW_LENGTH] = .NEW_MSG_LEN + OPCSK_COMHDRSIZ;
428 0425 REFORMAT_DESC [DSCSB_DTYPE] = 0;
429 0426 REFORMAT_DESC [DSCSB_CLASS] = 0;
430 0427 REFORMAT_DESC [DSCSA_POINTER] = REFORMAT_BUFFER;
431 0428
432 0429
433 0430 Call the new-message handler to finish processing the message.
434 0431
435 0432 GLOBAL STATUS [GBLSTS_K_OLD_FORMAT_MSG] = TRUE; ! Mark this as an old format msg
436 0433 REPLY_HANDLER (REFORMAT_DESC);
437 0434 GLOBAL STATUS [GBLSTS_K_OLD_FORMAT_MSG] = FALSE;
438 0435
439 0436 1 END; ! End of RPLY_HANDLER
```

				.EXTRN REPLY_HANDLER					
				SE	FSEC	01FC	00000	.ENTRY RPLY_HANDLER, Save R2,R3,R4,R5,R6,R7,R8	0302
				58	04	CE	9E	MOVAB -2580(SP), SP	
				33		AC	DO	MOVL BUFFER_DESC, R8	0368
						68	B1	CMPL (R8), #51	
						01	1E	BGEQU 18	
							04	RET	
							26	MOVC3 #38, 24(R8), REFORMAT_BUFFER	0375
09DA	BF	OC	AE	04			28	MOVC5 #0, (SP), #0, #2522, REFORMAT_BUFFER+38	0380
			00	6E			2C		
			52		32	AE			
				04					
				A8		26	C1	ADDL3 #38, 4(R8), OLD_MSG	0385
				56	32	AE	9E	MOVAB REFORMAT_BUFFER+38, NEW_MSG	0386
				66		62	90	MOVB (OLD_MSG), (NEW_MSG)	0387
			01	A6		01	90	MOVB #1, T(NEW_MSG)	0388
			12	A6	04	A2	DO	MOVL 4(OLD_MSG), 18(NEW_MSG)	0389
			06	A6	02	A2	3C	MOVZWL 2(OLD_MSG), 6(NEW_MSG)	0390
			08	A6		05	A0	ADDW2 #5, 8(NEW_MSG)	
			F8	AD		14	DO	MOVL #20, DEV_DESC	0396
			FC	AD	1B	A6	9E	MOVAB 27(R6), DEV_DESC+4	0397
				7E	08	A2	3C	MOVZWL 8(OLD_MSG), --(SP)	0398
					0A	A2	9F	PUSHAB 10(OLD_MSG)	
					F8	AD	9F	PUSHAB DEV_DESC	
					OC	AE	9F	PUSHAB OUT_LENGTH	
					0000G	CF	9F	PUSHAB DEVICE_FAO	
			00000000G	00		05	FB	CALLS #5, SYSSFAO	
			1A	A6		6E	90	MOVB OUT_LENGTH, 26(NEW_MSG)	0399
				50	40	8F	9A	MOVZBL #64, OLD_MSG_LEN	0400
				57		6E	3C	MOVZWL OUT_LENGTH, NEW_MSG_LEN	0401
				57		1B	CO	ADDL2 #27, NEW_MSG_LEN	
50		68		10		00	ED	CMPLV #0, #16, (R8), OLD_MSG_LEN	0405
						1C	15	BLEQ 28	
				58		68	3C	MOVZWL (R8), R8	0412

02	A746	1A	58	50	C2	00077	SUBL2	OLD_MSG_LEN, R8	:	0414
			A2	58	28	0007A	MOVC3	R8, -26(OLD_MSG), 2(NEW_MSG_LEN)[NEW_MSG]	:	0416
			9E	6746	9F	00081	PUSHAB	(NEW_MSG_LEN)[NEW_MSG]	:	0417
				58	B0	00084	MOVW	R8, 3(SPT+)	:	
				6746	9F	00087	PUSHAB	(NEW_MSG_LEN)[NEW_MSG]	:	0417
			50	9E	3C	0008A	MOVZWL	3(SPT+, R0)	:	
			57	50	C0	0008D	ADDL2	R0, NEW_MSG_LEN	:	0419
04	AE		57	02	C0	00090	ADDL2	#2, NEW_MSG_LEN	:	0424
			57	26	A1	00093	ADDW3	#38, NEW_MSG_LEN, REFORMAT_DESC	:	0425
		08	AE	06	AE	B4	CLRW	REFORMAT_DESC+2	:	0427
		0000G	CF	0C	AE	9E	MOVAB	REFORMAT_BUFFER, REFORMAT_DESC+4	:	0432
					10	88	BISB2	#16, GLOBAL_STATUS	:	0433
		0000G	CF	04	AE	9F	PUSHAB	REFORMAT_DESC	:	0434
		0000G	CF		01	FB	CALLS	#1, REPLY_HANDLER	:	0436
		0000G	CF		10	8A	BICB2	#16, GLOBAL_STATUS	:	
					04	000B2	RET		:	

; Routine Size: 179 bytes, Routine Base: \$CODE\$ + 0008


```
441 0437 1 GLOBAL ROUTINE RQST_HANDLER (BUFFER_DESC) : NOVALUE =
442 0438 1
443 0439 1
444 0440 1 **
445 0441 1 Functional description:
446 0442 1 This routine is the handler for all RQST messages received by OPCOM.
447 0443 1 This message is in the old format, and must be converted to the new
448 0444 1 format before it can be processed. Once this is done, the new format
449 0445 1 message handler is called to process the reformatted request.
450 0446 1
451 0447 1 Input:
452 0448 1
453 0449 1 BUFFER_DESC : The address of a quadword buffer descriptor that
454 0450 1 describes the buffer containing the message.
455 0451 1
456 0452 1 Implicit Input:
457 0453 1
458 0454 1 None.
459 0455 1
460 0456 1 Output:
461 0457 1
462 0458 1 None.
463 0459 1
464 0460 1 Implicit output:
465 0461 1
466 0462 1 Some accounting data will be updated
467 0463 1 to reflect the receipt of the message.
468 0464 1
469 0465 1 Side effects:
470 0466 1
471 0467 1 None.
472 0468 1
473 0469 1 Routine value:
474 0470 1
475 0471 1 None.
476 0472 1 --
477 0473 1
478 0474 2 BEGIN ! Start of RQST_HANDLER
479 0475 2
480 0476 2 MAP
481 0477 2
482 0478 2 BUFFER_DESC : $ref_bblock;
483 0479 2
484 0480 2 EXTERNAL
485 0481 2 GLOBAL_STATUS : BITVECTOR; ! Global status flags
486 0482 2
487 0483 2 EXTERNAL ROUTINE
488 0484 2 REQUEST_HANDLER : NOVALUE; ! New format msg handler
489 0485 2
490 0486 2 LOCAL
491 0487 2
492 0488 2 OLD_MSG_LEN : LONG, ! Length of old message
493 0489 2 OLD_MSG : $ref_bblock, ! Pointer to start of old message
494 0490 2 NEW_MSG : $ref_bblock, ! Pointer to start of new message
495 0491 2 REFORMAT_BUFFER : $bblock [OP($K_MAXREAD), ! Buffer to hold the reformatted message
496 0492 2 REFORMAT_DESC : $desc_block; ! Descriptor for the REFORMAT_BUFFER
497 0493 2
```

```

0498      0494      : Make sure the message is big enough. If not, it
0499      0495      : cannot possibly be a valid message, and OPCOM will
0500      0496      : simply ignore it.
0501      0497      :
0502      0498      : IF .BUFFER_DESC [DSC$W_LENGTH] LSS (OPC$K_COMHDRSIZ + 8)
0503      0499      THEN
0504      0500      RETURN;
0505      0501      :
0506      0502      : Copy the common header provided by $SENDPR to the new buffer
0507      0503      : CHSMOVE (OPC$K_COMHDRSIZ, .BUFFER_DESC [DSC$A_POINTER], REFORMAT_BUFFER);
0508      0504      :
0509      0505      : Zero the remainder of the REFORMAT_BUFFER.
0510      0506      : CHSFILL (0, (OPC$K_MAXREAD - OPC$K_COMHDRSIZ), (REFORMAT_BUFFER + OPC$K_COMHDRSIZ));
0511      0507      :
0512      0508      : Move the old message fields into their corresponding places in the new message format.
0513      0509      :
0514      0510      OLD_MSG = .BUFFER_DESC [DSC$A_POINTER] + OPC$K_COMHDRSIZ;
0515      0511      NEW_MSG = REFORMAT_BUFFER + OPC$K_COMHDRSIZ;
0516      0512      NEW_MSG [OPC$B_RQSTCODE] = .OLD_MSG [OPC$B_MS_TYPE];           ! Set message code
0517      0513      NEW_MSG [OPC$B_SCOPE] = OPC$K_SYSTEM;                         ! Force SYSTEM request
0518      0514      NEW_MSG [OPC$L_ATTNUMASK1] = .OLD_MSG [$BYTEOFFSET (OPC$B_MS_TARGET), 0, 24, 0];
0519      0515      NEW_MSG [OPC$L_RQSTID] = .OLD_MSG [OPC$L_MS_RQSTID];
0520      0516      OLD_MSG_LEN = OPC$K_COMHDRSIZ + $BYTEOFFSET (OPC$L_MS_TEXT);
0521      0517      :
0522      0518      : Copy the request text, if any, to the new message buffer.
0523      0519      :
0524      0520      IF .BUFFER_DESC [DSC$W_LENGTH] GTR .OLD_MSG_LEN
0525      0521      THEN
0526      0522      BEGIN
0527      0523      NEW_MSG [OPC$W_REQUEST_LENGTH] = .BUFFER_DESC [DSC$W_LENGTH] - .OLD_MSG_LEN;
0528      0524      CHSMOVE (.NEW_MSG [OPC$W_REQUEST_LENGTH], OLD_MSG [OPC$L_MS_TEXT], NEW_MSG [OPC$T_REQUEST_TEXT]);
0529      0525      END;
0530      0526      :
0531      0527      : Create a descriptor for the reformatted message.
0532      0528      :
0533      0529      REFORMAT_DESC [DSC$W_LENGTH] = OPC$K_COMHDRSIZ + OPC$K_HDR_SIZE + .NEW_MSG [OPC$W_REQUEST_LENGTH] + 2;
0534      0530      REFORMAT_DESC [DSC$B_DTYPE] = 0;
0535      0531      REFORMAT_DESC [DSC$B_CLASS] = 0;
0536      0532      REFORMAT_DESC [DSC$A_POINTER] = REFORMAT_BUFFER;
0537      0533      :
0538      0534      : Call the new-message handler to finish processing the message.
0539      0535      :
0540      0540      GLOBAL STATUS [GBLSTS_K_OLD_FORMAT_MSG] = TRUE;                ! Mark this as an old format msg
0541      0541      REQUEST_HANDLER (REFORMAT_DESC);
0542      0542      GLOBAL STATUS [GBLSTS_K_OLD_FORMAT_MSG] = FALSE;
0543      0543      :
0544      0544      END;
0545      0545      : End of RQST HANDLER
0546      0546
0547      0547
0548      0544
0549      0545
0550      0546
0551      0547
0552      0548

```

										.EXTRN REQUEST_HANDLER		
										.ENTRY	RQST_HANDLER, Save R2,R3,R4,R5,R6,R7	0437
										MOVAB	-2568(SP), SP	
										MOVL	BUFFER_DESC, R7	0499
										CMPW	(R7), #46	
										BLSSU	28	
										MOVC3	#38, 34(R7), REFORMAT_BUFFER	0506
										MOVC5	#0, (SP), #0, #2522, REFORMAT_BUFFER+38	0511
										ADDL3	#38, 4(R7), OLD_MSG	0516
										MOVAB	REFORMAT_BUFFER+38, NEW_MSG	0517
										MOVB	(OLD_MSG), (NEW_MSG)	0518
										MOVB	#1, T(NEW_MSG)	0519
										EXTZV	#0, #24, T(OLD_MSG), 10(NEW_MSG)	0520
										MOVL	4(OLD_MSG), 18(NEW_MSG)	0521
										MOVL	#46, OLD_MSG_LEN	0522
										CMPZV	#0, #16, (R7), OLD_MSG_LEN	0526
										BLEQ	18	
										SUBW3	OLD_MSG_LEN, (R7), 26(NEW_MSG)	0529
										MOVC3	26(NEW_MSG), 8(OLD_MSG), 28(NEW_MSG)	0530
										ADDW3	#66, 28(NEW_MSG), REFORMAT_DESC	0536
										CLRW	REFORMAT_DESC+2	0537
										MOVAB	REFORMAT_BUFFER, REFORMAT_DESC+4	0539
										BISB2	#16, GLOBAL_STATUS	0544
										PUSHL	SP	0545
										CALLS	#1, REQUEST_HANDLER	
										BICB2	#16, GLOBAL_STATUS	0546
										RET		0548

										00FC 00000	
										CE 9E 00002	
										AC D0 00007	
										67 B1 0000B	
										61 1F 0000E	
										26 28 00010	
										00 2C 00016	
										AE 0001D	
										26 C1 0001F	
										AE 9E 00024	
										60 90 00028	
										01 90 0002B	
										00 EF 0002F	
										A0 D0 00036	
										2E D0 00038	
										00 ED 0003E	
										0C 15 00043	
										51 A3 00045	
										A6 28 0004A	
										8F A1 00051	1\$:
										AE B4 00058	
										AE 9E 0005B	
										10 88 00060	
										5E DD 00065	
										01 FB 00067	
										10 8A 0006C	
										04 00071	2\$:

09DA	BF	08	AE	04	B7				
			00		6E				
			50	04	A7				
					56				
					66				
0A	A6	01	A0	01	A6				
				12	A6	04			
	51				51				
					10				
		1A	A6		67				
		1C	A6	08	A0	1A			
			6E	1A	A6	0042			
				04	AE	02			
				0000G	CF	08			
				0000G	CF				
				0000G	CF				

; Routine Size: 114 bytes, Routine Base: \$CODE\$ + 018E


```
554 0549 1 GLOBAL ROUTINE SECU_HANDLER (BUFFER_DESC) : NOVALUE =
555 0550 1
556 0551 1 ++
557 0552 1 Functional description:
558 0553 1
559 0554 1 This routine is the handler for all SECURITY messages received by OPCOM.
560 0555 1 This message is in the old format, and must be converted to the new
561 0556 1 format before it can be processed. Once this is done, the new format
562 0557 1 message handler is called to process the reformatted request.
563 0558 1
564 0559 1 Input:
565 0560 1
566 0561 1 BUFFER_DESC : The address of a quadword buffer descriptor that
567 0562 1 describes the buffer containing the message.
568 0563 1
569 0564 1 Implicit Input:
570 0565 1
571 0566 1 None.
572 0567 1
573 0568 1 Output:
574 0569 1
575 0570 1 None.
576 0571 1
577 0572 1 Implicit output:
578 0573 1
579 0574 1 Some accounting data will be updated
580 0575 1 to reflect the receipt of the message.
581 0576 1
582 0577 1 Side effects:
583 0578 1
584 0579 1 None.
585 0580 1
586 0581 1 Routine value:
587 0582 1
588 0583 1 None.
589 0584 1 --
590 0585 1
591 0586 2 BEGIN ! Start of SECU_HANDLER
592 0587 2
593 0588 2 MAP
594 0589 2
595 0590 2 BUFFER_DESC : $ref_bblock;
596 0591 2
597 0592 2 EXTERNAL
598 0593 2 GLOBAL_STATUS : BITVECTOR; ! Global status flags
599 0594 2
600 0595 2 EXTERNAL ROUTINE
601 0596 2 SECURITY_HANDLER : NOVALUE; ! New format msg handler
602 0597 2
603 0598 2 LOCAL
604 0599 2
605 0600 2 OLD_MSG_LEN : LONG, ! Length of old message
606 0601 2 OLD_MSG : $ref_bblock, ! Pointer to start of old message
607 0602 2 NEW_MSG : $ref_bblock, ! Pointer to start of new message
608 0603 2 REFORMAT_BUFFER : $bblock [OP($K_MAXREAD)], ! Buffer to hold the reformatted message
609 0604 2 REFORMAT_DESC : $desc_block; ! Descriptor for the REFORMAT_BUFFER
610 0605 2
```

[illegible]

				.EXTRN SECURITY_HANDLER						
				5E	F5F8	00FC	00000	.ENTRY	SECURITY_HANDLER, Save R2,R3,R4,R5,R6,R7	0549
				57	04	CE	9E	MOVAB	-2568(SP), SP	
				2E		AC	D0	MOVL	BUFFER_DESC, R7	0611
						67	B1	CMPL	(R7), #46	
						61	1F	BLSSU	25	
						26	28	MOVC3	#38, 34(R7), REFORMAT_BUFFER	0618
						00	2C	MOVC5	#0, (SP), #0, #2522, REFORMAT_BUFFER+38	0623
						AE	0001D			
						26	C1	ADDL3	#38, 4(R7), OLD_MSG	0628
						AE	9E	MOVAB	REFORMAT_BUFFER+38, NEW_MSG	0629
						60	90	MOVB	(OLD_MSG), (NEW_MSG)	0630
						01	90	MOVB	#1, T(NEW_MSG)	0631
						00	EF	EXTZV	#0, #24, T(OLD_MSG), 10(NEW_MSG)	0632
						A0	D0	MOVL	4(OLD_MSG), 18(NEW_MSG)	0633
						2E	D0	MOVL	#46, OLD_MSG_LEN	0634
						00	ED	CMPLV	#0, #16, (R7), OLD_MSG_LEN	0638
						0C	15	BLEQ	15	
						51	A3	SUBW3	OLD_MSG_LEN, (R7), 26(NEW_MSG)	0641
						A6	28	MOVC3	26(NEW_MSG), 8(OLD_MSG), 28(NEW_MSG)	0642
						8F	A1	ADDW3	#66, 28(NEW_MSG), REFORMAT_DESC	0648
						AE	B4	CLRW	REFORMAT_DESC+2	0649
						AE	9E	MOVAB	REFORMAT_BUFFER, REFORMAT_DESC+4	0651
						10	88	BISB2	#16, GLOBAL_STATUS	0656
						5E	DD	PUSHL	SP	0657
						01	FB	CALLS	#1, SECURITY_HANDLER	
						10	8A	BICB2	#16, GLOBAL_STATUS	0658
						04	00071	RET		0660

; Routine Size: 114 bytes, Routine Base: \$CODE\$ + 0200


```
667 0661 1 GLOBAL ROUTINE STS_HANDLER (BUFFER_DESC) : NOVALUE =
668 0662 1
669 0663 1 **
670 0664 1 Functional description:
671 0665 1
672 0666 1 This routine is the handler for all STS messages received by OPCOM.
673 0667 1 This message is in the old format, and must be converted to the new
674 0668 1 format before it can be processed. Once this is done, the new format
675 0669 1 message handler is called to process the reformatted request.
676 0670 1
677 0671 1 Input:
678 0672 1
679 0673 1 BUFFER_DESC : The address of a quadword buffer descriptor that
680 0674 1 describes the buffer containing the message.
681 0675 1
682 0676 1 Implicit Input:
683 0677 1
684 0678 1 None.
685 0679 1
686 0680 1 Output:
687 0681 1
688 0682 1 None.
689 0683 1
690 0684 1 Implicit output:
691 0685 1
692 0686 1 Some accounting data will be updated
693 0687 1 to reflect the receipt of the message.
694 0688 1
695 0689 1 Side effects:
696 0690 1
697 0691 1 None.
698 0692 1
699 0693 1 Routine value:
700 0694 1
701 0695 1 None.
702 0696 1 --
703 0697 1
704 0698 2 BEGIN ! Start of STS_HANDLER
705 0699 2
706 0700 2 MAP
707 0701 2
708 0702 2 BUFFER_DESC : $ref_bblock;
709 0703 2
710 0704 2 EXTERNAL
711 0705 2 GLOBAL_STATUS : BITVECTOR, ! Global status flags
712 0706 2 DEVICE_FAO : $bblock; ! FAO control string
713 0707 2
714 0708 2 EXTERNAL ROUTINE
715 0709 2 STATUS_HANDLER : NOVALUE; ! New format msg handler
716 0710 2
717 0711 2 LOCAL
718 0712 2
719 0713 2 OLD_MSG : $ref_bblock, ! Pointer to start of old message
720 0714 2 NEW_MSG : $ref_bblock, ! Pointer to start of new message
721 0715 2 OUT_LENGTH : WORD, ! Length of formatted operator device name
722 0716 2 DEV_DESC : $desc_block, ! Operator device name descriptor
723 0717 2 REFORMAT_BUFFER : $bblock [OP($K_MAXREAD), ! Buffer to hold the reformatted message
```

```
724      REFORMAT_DESC : $desc_block;                ! Descriptor for the REFORMAT_BUFFER
725
726
727      Make sure the message is big enough.  If not, it
728      cannot possibly be a valid message, and OPCOM will
729      simply ignore it.
730
731      IF .BUFFER_DESC [DSCSW_LENGTH] LSS (OPCSK_COMHDRSIZ + 7)
732      THEN
733          RETURN;
734
735      Copy the common header provided by $SENDPR to the new buffer
736      CH$MOVE (OPCSK_COMHDRSIZ, .BUFFER_DESC [DSCSA_POINTER], REFORMAT_BUFFER);
737
738      Zero the remainder of the REFORMAT_BUFFER.
739      CH$FILL (0, (OPCSK_MAXREAD - OPCSK_COMHDRSIZ), (REFORMAT_BUFFER + OPCSK_COMHDRSIZ));
740
741      Move the old message fields into their corresponding places in the new message format.
742      OLD_MSG = .BUFFER_DESC [DSCSA_POINTER] + OPCSK_COMHDRSIZ;
743      NEW_MSG = REFORMAT_BUFFER + OPCSK_COMHDRSIZ;
744      NEW_MSG [OPCSB_RSCODE] = .OLD_MSG [OPCSB_MS_TYPE];
745      NEW_MSG [OPCSB_SCOPE] = OPCSK_SYSTEM;
746
747      Build the operator device name from the ASCII device string
748      and the device unit number.  Build the FAO OUTBUF descriptor
749      to point to the correct spot within NEW_MSG to save a copy.
750
751      DEV_DESC [0,0,32,0] = 20;                                ! Allow for a large device name
752      DEV_DESC [DSCSA_POINTER] = .NEW_MSG + $BYTEOFFSET (OPCST_STATUS_OPR) + 1;
753      $FAO (DEVICE FAO, OUT_LENGTH, DEV_DESC, OLD_MSG [OPCST_MS_ONAME], .OLD_MSG [OPCSW_MS_OUNIT]);
754      NEW_MSG [$BYTEOFFSET TOPCST_STATUS_OPR, 0,8,0] = .OUT_LENGTH;
755
756
757      Create a descriptor for the reformatted message.
758
759      REFORMAT_DESC [DSCSW_LENGTH] = OPCSK_COMHDRSIZ + OPCSK_HDR_SIZE + .OUT_LENGTH + 1;
760      REFORMAT_DESC [DSCSB_DTYPE] = 0;
761      REFORMAT_DESC [DSCSB_CLASS] = 0;
762      REFORMAT_DESC [DSCSA_POINTER] = REFORMAT_BUFFER;
763
764      Call the new-message handler to finish processing the message.
765
766      GLOBAL_STATUS [GBLSTS_K_OLD_FORMAT_MSG] = TRUE;          ! Mark this as an old format msg
767      STATUS_HANDLER (REFORMAT_DESC);
768      GLOBAL_STATUS [GBLSTS_K_OLD_FORMAT_MSG] = FALSE;
769
770      END;
771
772      ! End of STS_HANDLER
```

.EXTRN STATUS_HANDLER

```

.ENTRY STS_HANDLER, Save R2,R3,R4,R5,R6      : 0661
MOVAB  -2580(SP), SP                          :
MOVL   BUFFER_DESC, R6                       : 0725
CMPW   (R6), #45                             :
BLSSU  1$                                     :
MOVCS  #38, 34(R6), REFORMAT_BUFFER         : 0732
MOVCS  #0, (SP), #0, #2522, REFORMAT_BUFFER+38 : 0737

ADDL3  #38, 4(R6), OLD_MSG                   : 0742
MOVAB  REFORMAT_BUFFER+38, NEW_MSG           : 0743
MOVB   (OLD_MSG), (NEW_MSG)                  : 0744
MOVB   #1, T(NEW_MSG)                        : 0745
MOVL   #20, DEV_DESC                         : 0751
MOVAB  27(R2), DEV_DESC+4                    : 0752
MOVZWL 8(OLD_MSG), -(SP)                     : 0753
PUSHAB 10(OLD_MSG)
PUSHAB DEV_DESC
PUSHAB OUT_LENGTH
PUSHAB DEVICE_FAO
CALLS  #5, SYS$FAO
MOVB   OUT_LENGTH, 26(NEW_MSG)               : 0754
ADDW3  #65, OUT_LENGTH, REFORMAT_DESC        : 0760
CLRW   REFORMAT_DESC+2                       : 0761
MOVAB  REFORMAT_BUFFER, REFORMAT_DESC+4      : 0763
BISB2  #16, GLOBAL_STATUS                    : 0768
PUSHAB REFORMAT_DESC                         : 0769
CALLS  #1, STATUS_HANDLER
BICB2  #16, GLOBAL_STATUS
RET                                           : 0770
                                           : 0772

```

```

09DA 8F      OC AE 04 B6 6E      F5EC CE 9E 00000
                    56 04 AC D0 00002
                    2D 66 B1 0000B
                    65 1F 0000E
                    26 28 00010
                    00 2C 00016
                    32 AE 0001D
                    26 C1 0001F
                    32 AE 9E 00024
                    60 90 00028
                    01 90 0002B
                    F8 AD 14 D0 0002F
                    FC AD 1B A2 9E 00033
                    7E 08 A0 3C 00038
                    0A A0 9F 0003C
                    F8 AD 9F 0003F
                    0C AE 9F 00042
                    0000G CF 9F 00045
                    05 FB 00049
                    00 6E 90 00050
                    04 AE 8F A1 00054
                    06 AE B4 0005B
                    0C AE 9E 0005E
                    08 AE 88 00063
                    0000G CF 04 AE 9F 00068
                    0000G CF 01 FB 0006B
                    0000G CF 10 8A 00070
                    04 04 00075 1$:

```

; Routine Size: 118 bytes, Routine Base: \$CODE\$ + 0272


```
0773 1 GLOBAL ROUTINE TERME_HANDLER (BUFFER_DESC) : NOVALUE =
0774 1
0775 1 ++
0776 1 Functional description:
0777 1
0778 1 This routine is the handler for all TERME messages received by OPCOM.
0779 1 This message is in the old format, and must be converted to the new
0780 1 format before it can be processed. Once this is done, the new format
0781 1 message handler is called to process the reformatted request.
0782 1
0783 1 Input:
0784 1
0785 1 BUFFER_DESC : The address of a quadword buffer descriptor that
0786 1 describes the buffer containing the message.
0787 1
0788 1 Implicit Input:
0789 1
0790 1 None.
0791 1
0792 1 Output:
0793 1
0794 1 None.
0795 1
0796 1 Implicit output:
0797 1
0798 1 Some accounting data will be updated
0799 1 to reflect the receipt of the message.
0800 1
0801 1 Side effects:
0802 1
0803 1 None.
0804 1
0805 1 Routine value:
0806 1
0807 1 None.
0808 1
0809 1 --
0810 1
0811 1 BEGIN ! Start of TERME_HANDLER
0812 1
0813 1 MAP
0814 1
0815 1 BUFFER_DESC : $ref_bblock;
0816 1
0817 1 EXTERNAL ROUTINE
0818 1 OPRENABLE_HANDLER : NOVALUE; ! New format message handler
0819 1
0820 1 EXTERNAL
0821 1 GLOBAL_STATUS : BITVECTOR, ! Global status flags
0822 1 DEVICE_FAO : $bblock; ! FAO control string
0823 1
0824 1 LOCAL
0825 1
0826 1 OLD_MSG : $ref_bblock, ! Pointer to start of old message
0827 1 NEW_MSG : $ref_bblock, ! Pointer to start of new message
0828 1 OUT_LENGTH : WORD, ! Length of formatted operator device name
0829 1 DEV_DESC : $desc_block, ! Operator device name descriptor
0830 1 REFORMAT_BUFFER : $bblock [OP($K_MAXREAD), ! Buffer to hold the reformatted message
0831 1
0832 1
0833 1
0834 1
0835 1
0836 1
```

```

837      0830      2      REFORMAT_DESC      : $desc_block;      ! Descriptor for the REFORMAT_BUFFER
838      0831
839      0832
840      0833      ! Make sure the message is big enough. If not, it
841      0834      ! cannot possibly be a valid message, and OPCOM will
842      0835      ! simply ignore it.
843      0836
844      0837      IF .BUFFER_DESC [DSC$W_LENGTH] LSS (OPC$K_COMHDRSIZ + 13)
845      0838      THEN
846      0839      RETURN;
847      0840
848      0841      ! Copy the common header provided by $SENDPR to the new buffer
849      0842
850      0843      CH$MOVE (OPC$K_COMHDRSIZ, .BUFFER_DESC [DSC$A_POINTER], REFORMAT_BUFFER);
851      0844
852      0845      ! Zero the remainder of the REFORMAT_BUFFER.
853      0846
854      0847      CH$FILL (0, (OPC$K_MAXREAD - OPC$K_COMHDRSIZ), (REFORMAT_BUFFER + OPC$K_COMHDRSIZ));
855      0848
856      0849
857      0850      ! Move the old message fields into their corresponding places in the new message format.
858      0851
859      0852      OLD_MSG = .BUFFER_DESC [DSC$A_POINTER] + OPC$K_COMHDRSIZ;      ! Set pointer to request text
860      0853      NEW_MSG = REFORMAT_BUFFER + OPC$K_COMHDRSIZ;      ! Set pointer to start of new message.
861      0854      NEW_MSG [OPC$B_RQSTCODE] = .OLD_MSG [OPC$B_MS_TYPE];      ! Set message type code
862      0855      NEW_MSG [OPC$B_SCOPE] = OPC$K_SYSTEM;      ! Force this to a SYSTEM enable
863      0856      $bblock [NEW_MSG [OPC$L_RQ_OPTIONS], OPC$V_PERMOPER] = TRUE;      ! Force this to a PERMANENT enable
864      0857      IF .OLD_MSG [$BYTEOFFSET (OPC$B_MS_ENAB), 0, 24, 0] EQL 0
865      0858      THEN
866      0859      BEGIN
867      0860      $bblock [NEW_MSG [OPC$L_RQ_OPTIONS], OPC$V_DISABLE] = TRUE; ! This is a DISABLE request
868      0861      $bblock [NEW_MSG [OPC$L_RQ_OPTIONS], OPC$V_PERMOPER] = FALSE; ! Clear the PERMOPR bit
869      0862      END;
870      0863      NEW_MSG [OPC$L_ATTNUMASK1] = .OLD_MSG [OPC$L_MS_MASK];      ! Set the enable/disable mask.
871      0864
872      0865      ! Create an operator device name from the device name ASCII
873      0866      ! string and the device unit #. Just assume that FAO succeeded.
874      0867      ! Set up the OUTBUF descriptor so it points to the correct spot
875      0868      ! in NEW_MSG. This will save a copy operation.
876      0869
877      0870      DEV_DESC [0, 0, 16, 0] = 20;      ! Allow for a large device name
878      0871      DEV_DESC [DSC$A_POINTER] = .NEW_MSG + $BYTEOFFSET (OPC$T_OPRENABLE_OPR) + 1;
879      0872      $FAO (DEVICE FAO, OUT_LENGTH, DEV_DESC, OLD_MSG [OPC$T_MS_ONAME], .OLD_MSG [OPC$W_MS_OUNIT]);
880      0873      NEW_MSG [$BYTEOFFSET (OPC$T_OPRENABLE_OPR), 0, 8, 0] = .OUT_LENGTH;
881      0874
882      0875
883      0876      ! Create a descriptor for the reformatted message.
884      0877
885      0878      REFORMAT_DESC [DSC$W_LENGTH] = OPC$K_COMHDRSIZ + OPC$K_HDR_SIZE + .OUT_LENGTH + 1;
886      0879      REFORMAT_DESC [DSC$B_DTYPE] = 0;
887      0880      REFORMAT_DESC [DSC$B_CLASS] = 0;
888      0881      REFORMAT_DESC [DSC$A_POINTER] = REFORMAT_BUFFER;
889      0882
890      0883
891      0884
892      0885
893      0886      ! Call the new-message handler to finish processing the message.
```

```
: 894      0887 2 !
: 895      0888 2 ! GLOBAL_STATUS [GBLSTS_K_OLD_FORMAT_MSG] = TRUE;      ! Mark this as an old format msg
: 896      0889 2 OPRENABLE_HANDLER (REFORMAT_DESC);
: 897      0890 2 GLOBAL_STATUS [GBLSTS_K_OLD_FORMAT_MSG] = FALSE;
: 898      0891 2
: 899      0892 1 END;      ! End of TERME_HANDLER
```

.EXTRN OPRENABLE_HANDLER

```
.ENTRY TERME_HANDLER, Save R2,R3,R4,R5,R6      : 0773
MOVAB -2580(SP), SP
MOVL BUFFER_DESC, R6      : 0837
CMPW (R6), #51
BLSSU 2$
MOVC3 #38, 24(R6), REFORMAT_BUFFER      : 0844
MOVC5 #0, (SP), #0, #2522, REFORMAT_BUFFER+38      : 0849

ADDL3 #38, 4(R6), OLD_MSG      : 0854
MOVAB REFORMAT_BUFFER+38, NEW_MSG      : 0855
MOVB (OLD_MSG), (NEW_MSG)      : 0856
MOVB #1, T(NEW_MSG)      : 0857
BISB2 #2, 6(NEW_MSG)      : 0858
CMPZV #0, #24, T(OLD_MSG), #0      : 0859
BNEQ 1$
BISB2 #1, 6(NEW_MSG)      : 0862
BICB2 #2, 6(NEW_MSG)      : 0863
MOVL 4(OLD_MSG), 10(NEW_MSG)      : 0865
MOVW #20, DEV_DESC      : 0872
MOVAB 27(R2), DEV_DESC+4      : 0873
MOVZWL 8(OLD_MSG), -(SP)      : 0874
PUSHAB 10(OLD_MSG)
PUSHAB DEV_DESC
PUSHAB OUT_LENGTH
PUSHAB DEVICE_FAO
CALLS #5, SY$FAO
MOVB OUT_LENGTH, 26(NEW_MSG)      : 0875
ADDW3 #65, OUT_LENGTH, REFORMAT_DESC      : 0880
CLRW REFORMAT_DESC+2      : 0881
MOVAB REFORMAT_BUFFER, REFORMAT_DESC+4      : 0883
BISB2 #16, GLOBAL_STATUS      : 0888
PUSHAB REFORMAT_DESC      : 0889
CALLS #1, OPRENABLE_HANDLER
BICB2 #16, GLOBAL_STATUS      : 0890
RET      : 0892
```

; Routine Size: 143 bytes, Routine Base: \$CODE\$ + 02E8

```
: 900      0893 1
: 901      0894 1 END
: 902      0895 0 ELUDOM
```

! End of OPCOMOLD

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	887	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	20	0	1000	00:01.9
\$255\$DUA28:[OPCOM.OBJ]OPCOMLIB.L32;1	633	24	3	43	00:00.9

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:OPCOMOLD/OBJ=OBJ\$:OPCOMOLD MSRC\$:OPCOMOLD/UPDATE=(ENH\$:OPCOMOLD)

: Size: 887 code + 0 data bytes
: Run Time: 00:22.0
: Elapsed Time: 01:12.2
: Lines/CPU Min: 2440
: Lexemes/CPU-Min: 20228
: Memory Used: 102 pages
: Compilation Complete

0290 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

